# Submission in Response to NSF CI 2030 Request for Information



**DATE AND TIME:** 2017-04-04 13:40:14
**REFERENCE NO:** 214



**Author Names & Affiliations**

- Harrison Prosper - Florida State University

**Contact Email Address (for NSF use only)**

(Hidden)

**Research Domain, discipline, and sub-discipline**

Experimental particle physics

**Title of Submission**

Assessing Uncertainty is the Key to the Future

**Abstract** (maximum ~200 words).

Many of us have had the following experience. During an annual physical, your doctor looks at a document filled with measurements of a variety of physiological attributes. She or he then makes a declaration about the general state of your health. However, none of the numbers come with assessments of their reliability, that is, there are no uncertainties. You are relying on the physician's years of experience to make a judgement based on these numbers and on what amounts to her or his knowledge of the time series of these numbers together with judgements based on a physical exam during the visit. Fast forward a quarter of a century from now when the doctor's assistant is an AI system that will calculate predictive probabilities of different health states. However, with all this sophistication, we shall be no better off then than we are today if the system fails to tell the doctor and the patient its degree of confidence in its predictions. Machine learning systems today tend to be categorical, this handwritten digit is a 9 and this one is an 8. They fail to quantify the reliability of their answers. We argue that this is a serious deficiency that needs to be addressed through a concerted research effort that will likely require a meeting of minds between different research communities.

**Question 1** Research Challenge(s) (maximum ~1200 words): Describe current or emerging science or engineering research challenge(s), providing context in terms of recent research activities and standing questions in the field.

There is a growing recognition that machine learning methods need to provide some measure of the reliability of their answers. This is particularly acute in what is the fastest growing sector of machine learning, the so-called deep learning, which has seen an explosive growth since the breakthroughs in 2006. At an abstract level deep learning is merely the use of function classes modeled as neural networks with multiple layers, of which there are now many varieties, for classification and regression problems. In a recent paper by Yarin Gal and Zoubin Ghahraman (https://arxiv.org/pdf/1506.02142.pdf, 2016), it is shown that dropout, the random elimination of nodes in a deep neural network (aka, a multi-layer neural network) during training, can be viewed as an approximate Bayesian inference procedure. In

principle, therefore, the highly developed theories of Bayesian inference could be brought to bear on the problem of assessing uncertainty. However, while the ideas of this paper represent a significant advance towards the goal of assessing uncertainty in the context of deep neural networks trained using the dropout algorithm, that paper is just one step in the right direction. There is a need for a broadly applicable, feasible, method of uncertainty assessment in machine learning systems. At one level, the problem in principle is straightforward. All methods, deep neural networks, boosted decision trees, support vector machines, etc., whether used for classification or regression, and whether each is just one element of a general AI system, is ultimately a function class with parameters that need to be fitted. If the problem to be addressed -- recognizing traffic signs, distinguishing between a benign or malignant tumor in a radiogram, or predicting the number of forest fires to be expected during the summer months in California -- can be construed as one based on a likelihood function, albeit one of enormous dimensions, either defined explicitly as an analytical function or implicitly as a vast cloud of points in the parameter space, then in principle the task at hand is to understand how to make the application of standard statistical theory to huge problems computationally feasible. If that problem could be solved, then standard statistical theory provides methods to assess uncertainty. The time seems right for a concerted effort to see whether, in principle, standard statistical theory can be used in machine learning to assess uncertainties, and, if so, how to do so in practice.

**Question 2** Cyberinfrastructure Needed to Address the Research Challenge(s) (maximum ~1200 words): Describe any limitations or absence of existing cyberinfrastructure, and/or specific technical advancements in cyberinfrastructure (e.g. advanced computing, data infrastructure, software infrastructure, applications, networking, cybersecurity), that must be addressed to accomplish the identified research challenge(s).

Today, graphics processing units (GPUs) are used to render the construction and application of, for example, deep neural networks almost routine. "Almost" because programming these systems still requires a level of expertise that is in relatively short supply. While several companies, e.g. NVIDIA, have invested huge sums to make their GPU products more user-friendly, there is still considerable way to go before the learning curve can be flattened sufficiently to capitalize on the availability of large numbers of people who know how to code. Hardware improvements continue apace and the demise of Moore's law is not yet in sight. The huge gap is, as always, in software. There is a need for advanced compilers that can take code in any existing, and future, computing language, e.g., Python, and automatically map problems to the most efficient implementation on the underlying massively parallel computational architecture (MPCA); today it is GPUs, tomorrow perhaps adaptive MPCAs. It is still necessary today to be cognizant of the underlying GPU in order to make the best use of it. And, it is necessary to have some understanding of what these GPUs at good at and what they are not good at. This is unfortunate because our greatest treasure by far is our time and our imagination. The tedious task of mapping problems to an MPCA should be left to our AI assistants. Here is a concrete example of such a problem.

Today, Markov chain Monte Carlo (MCMC) methods are the method of choice for ferociously difficult problems, such as computing high-dimensional integrals. However, MCMC is notoriously slow and convergence diagnostics are still more an art than a science. In a large-scale Bayesian calculation, one has likelihood p(x|q) and a prior prior(q), where x are the data, which could number in the millions (and one day in the billions), and q are the parameters, which could number in the tens of millions. The simplest constructive way to integrate over the parameters, in order for example to calculate a predictive probability, is to sample from the prior, which is typically a function from which it is easy to sample, e.g., the product of ten million normal densities. The integral is typically over some function f(y, q) and is approximated as the weighted sum over f(y, q), where the weight is the likelihood p(x|q). The severe problem with this method is that the support of the likelihood may be quite different from that of the prior. If so, only a small fraction of the sampled parameter points will give a useful contribution to the integral. But, we would not care if it were possible to sample rapidly say a billion points from the parameters using some MPCA and if we were able to salvage say 10,000 to 100,000 useful points. Writing code to do this using existing languages is straightforward, but doing so to make optimal use of an MPCA, such as a GPU is not. This is where much work is needed. In addition, work is needed to build nested hierarchies of MPCAs. Many calculations entail loops within loops. Suppose it were possible to farm out the billion samplings to an MPCA, the parallelism can be taken further. Since we may have millions of parameters, each sampled point will require sampling a million individual parameters. Ideally, the hardware would know how to reconfigure itself to map its available computational nodes to match the double-loop structure of the problem. Or perhaps this would occur through a combination of advanced compiler technology and hardware.

**Question 3** Other considerations (maximum ~1200 words, optional): Any other relevant aspects, such as organization, process, learning and workforce development, access, and sustainability, that need to be addressed; or any other issues that NSF should consider.

The NSF is a marvelous organization for fostering innovation. However, it is highly success oriented, by which I mean this: bold serious efforts that fail are not rewarded as much as more cautious serious efforts that succeed. This is understandable: the NSF disburses from the public purse and is accordingly accountable to the public as, of course, it should be. But, if we truly want to encourage "out-of-the-box" -- even slightly off the wall crazy -- thinking, we need to accept that most ideas of that ilk fail, not only because they may turn out to be impossible, but because they may turn out to be too difficult given current means. Somehow, the NSF needs to have a mechanism to "let a thousand flowers bloom", while recognizing and accepting that most flowers will wilt. But the few that remain standing could transform the world, and, if we are fortunate and attentive, for the better.